CST1
COMPUTER SCIENCE TRIPOS Part IB

Wednesday 5 June 2024 13:30 to 16:30

COMPUTER SCIENCE Paper 6

*Answer **five** questions.*

*Submit the answers in five **separate** bundles, each with its own cover sheet. On each cover sheet, write the numbers of **all** attempted questions, and circle the number of the question attached.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
Approved calculator permitted

1 Complexity Theory

Let Factor be the *decision* problem where given a pair of integers (x, k), the goal is to decide whether x has a factor smaller than k. Let Factoring be the *search* problem, where given an integer x, the goal is to output a prime factorisation of x. (In the following, carefully note the distinction between Factor and Factoring.)

(a) Prove that $\mathsf{Factor} \in \mathsf{NP} \cap \mathsf{coNP}$. [5 marks]

(b) Prove that if $\mathsf{P} = \mathsf{NP} \cap \mathsf{coNP}$, then there exists a polynomial-time algorithm for Factoring. [7 marks]

(c) Define the class BQP. Is $\mathsf{Factoring} \in \mathsf{BQP}$? [4 marks]

(d) Show that a quantum (BQP) algorithm for a problem P, which is correct with probability $2/3$ over the measurement, can be amplified into a quantum algorithm for P, which is correct with probability $1-o(1)$ over the measurement. [4 marks]

2 Complexity Theory

(a) Provide a precise definition of the following complexity classes: EXP, NEXP, and NPSPACE. [3 marks]

(b) Prove that $\mathsf{NPSPACE} \subseteq \mathsf{EXP} \subseteq \mathsf{NEXP}$. [7 marks]

(c) Prove that if $\mathsf{P} = \mathsf{NP}$, then $\mathsf{EXP} = \mathsf{NEXP}$. [10 marks]

3 Computation Theory

(*a*) Define the collection of *primitive recursive functions*. [2 marks]

(*b*) What does it mean to say that a function $f : \mathbb{N} \to \mathbb{N}$ is λ-*definable*. [2 marks]

(*c*) For each of these functions, show that it is primitive recursive and explain why it is λ-definable.

(*i*) $\text{fact}(x) \triangleq x!$; [7 marks]

(*ii*) $\text{cond}(x, y, z) \triangleq \begin{cases} y & \text{if } x = 0 \\ z & \text{otherwise} \end{cases}$ [7 marks]

(*d*) Give an example of a function that is λ-definable but not primitive recursive. You do not need to give a proof of the fact. [2 marks]

4 Computation Theory

Let P be any register machine program.

(*a*) What is the partial function $f\colon \mathbb{N} \rightharpoonup \mathbb{N}$ of one argument computed by P? [2 marks]

(*b*) What is the partial function $g\colon \mathbb{N}^2 \rightharpoonup \mathbb{N}$ of two arguments computed by P? [2 marks]

(*c*) Describe the construction of a *Gödel numbering* of register machine programs. That is, a bijection G between the natural numbers $\mathbb{N}$ and the collection of register machine programs. [5 marks]

We now write $\phi_i : \mathbb{N} \rightharpoonup \mathbb{N}$ for the partial function of one argument that is computed by the register machine program $G(i)$, and $\psi_i : \mathbb{N}^2 \rightharpoonup \mathbb{N}$ for the partial function of two arguments that is computed by the register machine program $G(i)$, where G is the Gödel numbering constructed in part (*c*).

(*d*) Show that the partial function $u\colon \mathbb{N}^2 \rightharpoonup \mathbb{N}$ defined by $u(i, x) = \phi_i(x)$ for all i and x is computable. You may assume standard results about register machine programs, as long as you state them in full and clearly. [5 marks]

(*e*) Sketch a proof to show that there is a computable partial function $s\colon \mathbb{N}^2 \rightharpoonup \mathbb{N}$ such that, for all $x, y, z, \in \mathbb{N}$:

$$\phi_{s(x,y)}(z) = \psi_x(y, z).$$

[6 marks]

5 Data Science

In order to judge whether early prehistoric humans had some form of religious belief, archaeologists have proposed studying the alignment of graves in a burial site. The thinking is that if the graves all point in the same direction then the society must have had some form of supernatural belief which led them to align the graves in this way. However, we wouldn't expect perfect alignment: the skeletal remains might be at an angle to the original grave; or there might be a mix of aligned and non-aligned graves from different waves of occupants.

You have been asked by an archaeologist friend to conduct a hypothesis test of whether a collection of graves shows alignment. The dataset is a list $[x_1, \ldots, x_n]$ of angles, $0 \leq x_i < 360$ where 0 denotes North, 90 denotes East, and so on.

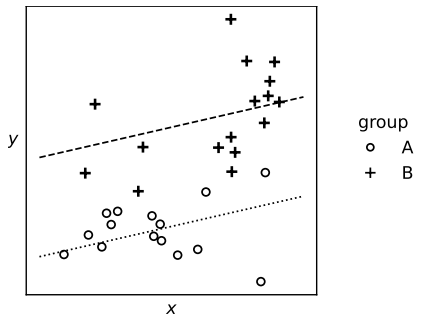
(*a*) What is meant by a "null hypothesis"? What null hypothesis do you propose? [5 marks]

(*b*) What is meant by a "test statistic"? What test statistic do you propose? Explain how your test statistic deals with the fact that 359.999 and 0 are very similar angles. [7 marks]

(*c*) What is meant by a "one-sided" or "two-sided" test? Which will you use? Explain your reasoning. [5 marks]

(*d*) Define "p-value". Explain how to compute the p-value for your test. [3 marks]

6 Data Science



This plot shows a collection of datapoints $(x_i, y_i, group_i)$ from two groups, together with fitted lines for each group. Based on our scientific understanding of the dataset, the lines should be parallel.

(*a*) Describe a probability model and a fitting procedure that might be behind this plot. Explain how you enforce the 'parallel lines' requirement. Give pseudocode for the fitting procedure. Explain briefly how the plot is produced. [*Note:* When describing a probability model, you should state your assumptions: what probability distributions are you using, and why have you chosen them?]
[7 marks]

(*b*) We would like to report our confidence about the offset between the two groups. Explain the Bayesian approach to computing this confidence, and give pseudocode. [7 marks]

(*c*) We have been asked to predict y at some new value x^*, for both groups. Explain how to compute confidence intervals for these two predictions, and give pseudocode. [6 marks]

7 Logic and Proof

A client wants a dispensing machine that sells two products, A and B. The following are some requirements:

1. In the *default state* a screen displays the two unselected products and it assumes no payment has been made.

2. Every time the machine is in the default state the user may select one and only one product.

3. It is always true that, given a selection, the user may pay for the selected product.

4. The selection can be changed provided that the user has not paid.

5. It is always possible to return to the default state.

Once a product has been selected and paid for, the user gets the product.

(a) Formalise statements 1 to 5 in S4 modal logic. Use letter A to represent the proposition that A is selected, B to represent that B is selected, D to represent the default state and P that a payment has been made. [6 marks]

(b) Model the default state as a world in a modal frame. Exhibit an interpretation that satisfies statements 1 to 5. [7 marks]

(c) Given your formalisation of statements 1 to 5, how would you formalise and prove (using the sequent calculus for S4) that, starting from the default state, it is possible for the user to get one of every product? Your answer can be schematic. A full formal proof is not necessary, but you must be clear about the strategy used. [*Hint:* Start by using the default state and statement 2 to prove that it is possible to select A.] [7 marks]

8 Logic and Proof

(a) Consider the following formulae, where a, b and c are constants and v, w, x, y and z are variables:

$$P(a, v) \rightarrow \neg Q(b, w) \tag{1}$$
$$\neg(\neg Q(b, x) \wedge P(b, y)) \tag{2}$$
$$\neg(\neg P(z, z) \wedge \neg P(z, c)) \tag{3}$$

(i) Convert the formulae above into conjunctive normal form (CNF) and express the result as a set of clauses. State which rule you used for each conversion step. [3 marks]

(ii) Convert the clauses resulting from part (a)(i) into Kowalski form. [2 marks]

(iii) Using the clauses resulting from part (a)(ii), give a resolution proof for them. Use clause (1) as the top clause. Indicate the selected literal(s), clause and substitution used at each step. [7 marks]

(b) (i) Convert the following formulae into clauses:

$$M \rightarrow (N \rightarrow M) \tag{4}$$
$$M \rightarrow (N \vee P) \tag{5}$$
$$N \rightarrow (\neg Q \wedge \neg R) \tag{6}$$
$$P \rightarrow (R \wedge \neg Q) \tag{7}$$
$$M \tag{8}$$

[2 marks]

(ii) Use the DPLL method to find a model satisfying the clauses from part (b)(i), or to prove that no such model exists. Briefly explain your work in each step. [6 marks]

9 Semantics of Programming Languages

(a) Suppose the types $\mathsf{InputChannel}(\tau)$, $\mathsf{OutputChannel}(\tau)$, and $\mathsf{IOChannel}(\tau)$ have the following API:

$$\begin{array}{lcl} \mathsf{read} & : & \mathsf{InputChannel}(\tau) \to \tau \\ \\ \mathsf{write} & : & \mathsf{OutputChannel}(\tau) \times \tau \to \mathsf{unit} \\ \\ \mathsf{read} & : & \mathsf{IOChannel}(\tau) \to \tau \\ \mathsf{write} & : & \mathsf{IOChannel}(\tau) \times \tau \to \mathsf{unit} \end{array}$$

Note that read and write are overloaded functions. Define a suitable subtyping relation over the channel types. [5 marks]

(b) Consider the following two concurrent L1 programs:

Program 1: $\mathsf{r} := !\mathsf{r} + 1; \mathsf{r} := !\mathsf{r} + 1$

Program 2: $\mathsf{r} := !\mathsf{r} + 2$

Are these two programs semantically equivalent in a concurrent setting? Give an informal but precise argument if they are, or give a counterexample if not. [2 marks]

(c) Suppose we try to introduce a safe file-handling API in a language with higher-order functions and state (such as L3) by introducing the function $\mathsf{withFile} : (\mathsf{File} \to \mathsf{unit}) \to \mathsf{unit}$. This function creates a file, passes the file to its callback, and then closes the file after the callback returns. Is there any way for a File object to outlive the callback invocation and leak into the environment? Either argue that the API is safe, or give a counterexample. [3 marks]

(d) We can define the prefix relation $xs \sqsubseteq ys$ on lists as follows:

$$\frac{}{[] \sqsubseteq ys} \qquad \frac{xs \sqsubseteq ys}{x :: xs \sqsubseteq x :: ys}$$

(i) Prove that the prefix relation is reflexive. [3 marks]

(ii) Prove that the prefix relation is transitive. Inversion properties may be used without proof, as long as they are explicitly indicated. [7 marks]

10 Semantics of Programming Languages

Regular expressions are defined by the following grammar:

r	$::=$	c	Matches the single-character word c
	$\mid$	ϵ	Matches the empty word
	$\mid$	$r_1 \circ r_2$	Matches the concatenation of an r_1-word and an r_2-word
	$\mid$	0	Matches no words
	$\mid$	$r_1 + r_2$	Matches any r_1-word or r_2-word
	$\mid$	$r*$	Matches the concatenation of a finite number of r-words

(*a*) Give a set of inference rules defining a relation for when a word w is matched by a regular expression r. Use the notation $w \cdot w'$ to denote concatenation.

[8 marks]

(*b*) (*i*) Using the matching relation defined above, define a suitable notion of semantic equivalence $r_1 \simeq r_2$ for regular expressions. [4 marks]

(*ii*) Use this definition to prove that $(r + r') \simeq (r' + r)$. You may use inversion lemmas without proof, as long as they are explicitly indicated. [4 marks]

(*c*) Define an inductive relation r null characterizing the regular expressions r for which ϵ in r. [4 marks]

END OF PAPER